Fine-Grained Argument Unit Recognition and Classification

Dietrich Trautmann,[†] Johannes Daxenberger,[‡] Christian Stab,[‡] Hinrich Schütze,[†] Iryna Gurevych[‡]
[†]Center for Information and Language Processing (CIS), LMU Munich, Germany
[‡]Ubiquitous Knowledge Processing Lab (UKP-TUDA), TU Darmstadt, Germany
dietrich@trautmann.me; inquiries@cislmu.org
http://www.ukp.tu-darmstadt.de

Abstract

Prior work has commonly defined argument retrieval from heterogeneous document collections as a sentence-level classification task. Consequently, argument retrieval suffers both from low recall and from sentence segmentation errors making it difficult for humans and machines to consume the arguments. In this work, we argue that the task should be performed on a more fine-grained level of sequence labeling. For this, we define the task as Argument Unit Recognition and Classification (AURC). We present a dataset of arguments from heterogeneous sources annotated as *spans of tokens* within a sentence, as well as with a corresponding stance. We show that and how such difficult argument annotations can be effectively collected through crowdsourcing with high inter-annotator agreement. The new benchmark, AURC-8, contains up to 15% more arguments per topic as compared to annotations on the sentence level. We identify a number of methods targeted at AURC sequence labeling, achieving close to human performance on known domains. Further analysis also reveals that, contrary to previous approaches, our methods are more robust against sentence segmentation errors. We publicly release our code and the AURC-8 dataset.[1]

1 Introduction

Argumentation and reasoning are fundamental human skills. They play a major role in education, daily conversations, as well as in many professional contexts including journalism, politics and the law. Argumentative skills are trained, for example, in the context of (public) debates, which are an essential part of democratic societies. Argument mining (AM), i.e., the processing of argumentative structures in natural language using automatic methods (Peldszus and Stede 2013), has recently gained considerable attention in the AI community (Cabrio and Villata 2018; Lippi and Torroni 2016; Nguyen and Litman 2018). AM requires sophisticated reasoning about controversial subject matters well beyond mere syntactic or semantic understanding. In recent research on AM, two radically different paradigms have evolved: closed-domain *discourse-level* AM seeks to identify the argumentative structure of a debate or an argumentative text (e.g., a student essay). In contrast, *information-seeking* AM aims to identify self-contained argumentative statements relevant to the given topic from any source. The goal of this kind of AM is to identify a broad and diverse set of arguments, ideally reflecting different viewpoints and aspects of the topic of interest. Most approaches to automatic discourse-level AM (Wyner et al. 2010; Stab and Gurevych 2014) rely on the *claim-premise* model, which refers to the basic units of an argument as a controversial proposition (claim) and connected evidence (premise). The claim-premise model is the basis for the analysis of complex argumentative texts, which can be solved by AI-based automatic methods with success (Kuribayashi et al. 2019).

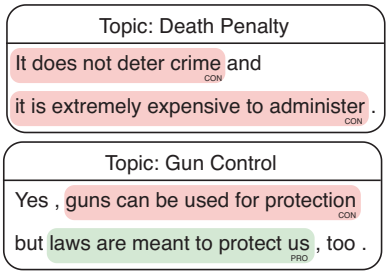

Figure 1: Annotation of argumentative spans and stance. Each of the two sentences contains two arguments.

The claim-premise model is, however, hardly applicable to text that does not contain an explicit argumentative structure (i.e., the majority of textual data) and, thus, has not been applied with much success to heterogeneous document collections (Habernal and Gurevych 2017). Information-seeking AM, which allows to detect arguments in sources that are not inherently argumentative, has been suggested as a remedy to this problem. It solves the following task: given a controversial claim ("topic"), detect supporting or opposing statements from (potentially) relevant sources. In this context, an argument is usually defined as a short text – or *span* – providing evidence or reasoning about the topic, supporting or opposing it (Stab et al. 2018b). This can be seen as a flat version of the claim-premise model, where the topic

[1]https://github.com/trtm/AURC

#	topic	#sentences	#candidates	#final	#arg-sent	#arg-unit	increase in %	#non-arg
T1	abortion	39,083	3,282	1,000	424	458	+8.02	576
T2	cloning	30,504	2,594	1,000	353	380	+7.65	647
T3	marijuana legalization	45,644	6,351	1,000	630	689	+9.37	370
T4	minimum wage	43,128	8,290	1,000	630	703	+11.59	370
T5	nuclear energy	43,576	5,056	1,000	623	684	+9.79	377
T6	death penalty	32,253	6,079	1,000	598	651	+8.86	402
T7	gun control	38,443	4,576	1,000	529	587	+10.96	471
T8	school uniforms	40,937	3,526	1,000	713	821	+15.15	287
	total	314,568	39,754	8,000	4,500	4,973	+10.51	3,500

Table 1: Sentences in the selection process and final corpus size. #arg-sent: argumentative sentences. #arg-unit: argumentative units. increase in %: increase of #arg-unit compared to #arg-sent.

is an implicit claim and the argument is a premise that supports or attacks the claim. Previous studies on information-seeking AM working with heterogeneous document collections have restricted arguments to be sentences. This assumption has been partly justified by the difficulty of "unitizing", i.e., of segmenting a sentence into meaningful units for argumentation tasks (Stab et al. 2018b). A few studies also allowed free text fragments as arguments (Aharoni et al. 2014), however, their arguments are extracted from very restricted domains (Wikipedia articles) with clear relevance to the topic. Consequently, it remains unclear whether such argument spans can be extracted from large and heterogeneous document collections with sufficient accuracy.

Information-seeking AM as defined above has thus not been applied to both highly heterogeneous document collections *and* on the token level. By identifying subsentence token spans as arguments for given topics in a large web crawl, we show that this task is feasible. Furthermore, addressing the unitizing problem, we show how the required training data can be created in a scalable manner and with high agreement using crowdsourcing. We call this task *Argument Unit Recognition and Classification (AURC)*.[2] Labeling arguments on the token level has several advantages. First and foremost, it prevents merging otherwise separate arguments into a single argument (e.g., for the topic *death penalty* in Figure 1), enabling us to retrieve a larger number of arguments (up to 15% more in our data). Second, it can handle two-sided argumentation in a single sentence adequately (e.g., for the topic *gun control* in Figure 1). It is also more robust against errors in sentence segmentation. Finally, downstream applications like fact checking that need to reason over a set of evidences require decomposing complex sentences into simpler argument units. Walton and Gordon (2017) argue that any kind of argument mining in which parts of a given text are re-used to form new arguments is closely related to argument invention (Levy et al. 2014). This view points towards further applications of our approach in the field of rhetoric.

We make the following contributions in this paper. First, we propose a slot-filling approach to argument annotation and show that it effectively collects token-level annotations through crowdsourcing with high inter-annotator agreement. Based on this approach, we construct AURC-8 by applying a sampling strategy to a large web crawl. Second, we present a number of methods for solving the AURC task and evaluate them on the AURC benchmark. Finally, we show that our token-level model does not depend on correct sentence boundary identification.

2 Related Work

Previous work on AM in AI and Natural Language Processing divides into discourse-level AM (Palau and Moens 2009; Stab and Gurevych 2014) and information-seeking AM (Shnarch et al. 2018; Wachsmuth et al. 2017; Hua and Wang 2017; Stab et al. 2018b), as explained in Section 1. Our work is in line with the latter: we model arguments as self-contained pieces of information that can be verified as relevant arguments for a given topic with no or minimal surrounding context. As opposed to previous work on information-seeking AM that extracted token-level arguments from Wikipedia articles (Levy et al. 2014), we apply our annotation schema and experiments to a much broader and noisier collection of web documents.

Ajjour et al. (2017) compare argumentative unit segmentation approaches on the token level across three corpora. They use a feature-based approach and various architectures for segmentation and find that BiLSTMs work best on average. In contrast to our work, they study argumentation from the discourse-level perspective, i.e., they do not consider topic-dependency and do not account for argument stance. Peldszus and Stede (2015) use elementary discourse units as starting point to classify argumentative discourse units (ADUs). As opposed to their work, our approach does not rely on discourse parsing to identify ADUs. Furthermore, Peldszus and Stede (2015) do not consider topic-dependency of arguments.

Multiple previous studies have shown that annotating arguments from the discourse-level perspective is very challenging for heterogeneous document collections. Habernal and Gurevych (2017) used Toulmin's schema of argumenta-

[2] A reviewer pointed out that neither argument unit recognition nor argument unit classification are novel subtasks and so the need for introducing a new abbreviation for their combination may be questioned. However, we are interested in the specific scenario of information-seeking AM for highly heterogeneous collections and for arguments segmented on the token level (i.e., fine-grained segmentation). This (in our view) novel scenario justifies a new abbreviation and a new name.

tion on several text genres including web data, resulting in rather low inter-annotator agreement scores (Krippendorff's α_u = 0.48). Miller, Sukhareva, and Gurevych (2019) apply the claim-premise model to customer reviews with similarly low agreement (α_u roughly between 0.4 and 0.5). Using the information-seeking perspective, Stab et al. (2018b) prove that arguments can be annotated with sufficient agreement in heterogeneous sources on the sentence level (Fleiss' κ=0.72). In our work, we propose a slot filling approach to transfer their findings to token-level arguments. Previously, Reisert et al. (2018) achieved good agreement determining complex reasoning structures with a set of highly specialized slots referred to as *argument templates*. Compared to their work, our slot filling approach is simpler, but it is applicable to heterogeneous sources and topics in a crowdsourcing environment.

3 Corpus Creation

Collecting annotations on the token level is challenging. First, the unit of annotation needs to be clearly defined. This is straightforward for tasks with short spans (sequences of words) such as named entities, but much harder for longer spans – as in the case of argument units. Second, labels from multiple annotators need to be merged into a single gold standard.[3] This is also more difficult for longer sequences because simple majority voting over individual words will likely create invalid (e.g., discontinuous or grammatically incorrect) spans. In the following, we propose solutions to these problems and describe the selection of sources, sampling and annotation for our novel argument unit dataset, AURC-8.

3.1 Data Source

We used the February 2016 Common Crawl archive,[4] which was indexed with Elasticsearch[5] following Stab et al. (2018a). Since we want to know, whether knowledge transfer to unknown topics is possible with few annotated topics, we limited the selection to Stab et al. (2018b)'s eight topics (cf. Table 1). This also increases comparability with related work. The topics are general enough to have good coverage in Common Crawl. They are also controversial and hence a good choice for argument mining with an expected diverse set of supporting and opposing arguments.

3.2 Retrieval Pipeline

For document retrieval, we queried the indexed data for Stab et al. (2018b)'s topics and collected the first 500 results per topic ordered by their document score (*doc_score*) from Elasticsearch; a higher *doc_score* indicates higher relevance for the topic. For each document, we retrieved the corresponding WARC file at the Common Crawl Index.[6] From there, we downloaded and parsed the original HTML document for the next steps of our pipeline; this ensures reproducibility. Following this, we used justext[7] to remove HTML boilerplate. We used spacy[8] to segment the resulting document into sentences and sentences into tokens. We only consider sentences with the number of tokens in the range $[3, 45]$.

3.3 Sentence Sampling

We pre-classified the selected sentences with a sentence-level argument mining model following Stab et al. (2018a) and available via the ArgumenText Classify API.[9] The API returns for each sentence (i) an argument confidence score *arg_score* in $[0.0, 1.0)$ (we discard sentences with *arg_score*< 0.5), (ii) the stance on the sentence level (PRO or CON) and (iii) the stance confidence score *stance_score*. This information was used together with the *doc_score* to rank sentences for a selection in the following crowd annotation process. See Table 1 for a detailed overview of the number of extracted sentences, as well as the number of candidates for the following ranking approach. We convert scores for documents, arguments and stance into ranks d_i, a_i and s_i. We then sum the three ranks to get an aggregated score for each sentence: $\text{agg}_i = d_i + a_i + s_i$. Sentences are divided by topic and pre-classified stance and ranked according to agg_i, for each combination separately. We then go down the ranked list selecting each sentence with probability 0.5 until the target size of $n = 500$ per stance and topic is reached; otherwise we do additional passes through the list. We adopted this probabilistic selection to ensure diversity – otherwise a long document with high relevance score at the top of the list might dominate the dataset with its sentences. Table 1 gives the final dataset creation statistics.

3.4 Crowd Annotations

Our goal is to develop a scalable approach to annotate argument units on the token level. Given that arguments need to be annotated with regard to a specific topic, training data for different topics needs to be created. As has been shown by previous work on information-seeking AM (Shnarch et al. 2018; Stab et al. 2018b), crowdsourcing (on the sentence level) can be used to obtain reliable annotations for argument mining datasets. However, as outlined above, token-level annotation significantly increases the difficulty of the annotation task. We distinguish between PRO (supporting) and CON (opposing) arguments and we use NON for non-argumentative text spans. Thus, we have a sequence labeling task with three classes: PRO, CON and NON. Our main question was: can we achieve sufficiently high agreement among untrained crowd workers for this task?

We use the α_u **agreement measure** (Krippendorff et al. 2016) in this work. It is designed for annotation tasks that

[3] One could also learn from "soft" labels, i.e., a distribution created from the votes of multiple annotators. However, this does not solve the problem that some annotators deliver low quality work and their votes should be outscored by a (hopefully) higher-quality majority of annotators.

[4] http://commoncrawl.org/2016/02/february-2016-crawl-archive-now-available/

[5] https://www.elastic.co/products/elasticsearch

[6] http://index.commoncrawl.org/CC-MAIN-2016-07

[7] http://corpus.tools/wiki/Justext

[8] https://spacy.io/

[9] https://api.argumentsearch.com

involve unitizing textual continua – i.e., segmenting continuous text into meaningful sub-units – and measuring chance-corrected agreement in those tasks. It is also a good fit for argument spans within a sentence: typically these spans are long and the context is a single sentence that may contain any type of argument and any number of arguments. Krippendorff et al. (2016) define a family of α-reliability coefficients that improve upon several weaknesses of previous α measures. From these, we chose the $\alpha_{u_{nom}}$ coefficient, which also takes into account agreement on "blanks" (non-arguments in our case) – since agreement on non-argument spans (including sentences without any arguments) is important as well and should not be ignored by the measure.

To determine agreement, we initially carried out an **in-house expert study** with three graduate employees (who were trained on the task beforehand) and randomly sampled 160 sentences (10 per topic and stance) from the overall data. In the first round, we did not impose any restrictions on the span of words to be selected, other than that the selected spans should be the shortest self-contained spans that form an argument. This resulted in unsatisfactory agreement ($\alpha_{u_{nom}}$ = 0.51, average over topics), one reason being inconsistency in selecting argument spans (median length of arguments ranged from nine to 16 words among the three experts).

In a second round, we therefore decided to restrict the spans that could be selected by applying a slot filling approach enforcing valid argument spans that match a template. We use the template: "<*TOPIC*> should be supported/opposed, because <*argument span*>". The guidelines specify that the resulting sentence must be grammatically correct.[10] Although this new setup increased the length of spans and reduced the total number of arguments selected, it increased consistency of spans substantially (min and max median lengths were now 15 and 17). Furthermore, the agreement between the three experts rose to $\alpha_{u_{nom}}$ = 0.61 (average over topics). Compared to other studies on token-level argument mining (Eckle-Kohler, Kluge, and Gurevych 2015; Li et al. 2017; Stab and Gurevych 2014), this score is in an acceptable range and we deem it sufficient to proceed with crowdsourcing. The averaged macro-F_1 over all pairs of expert annotations is 0.76 (referred to as human performance in Table 2).

In our **crowdsourcing setup**, workers could select one or multiple spans, where each span's permissible length is between one token and the entire sentence. Workers had to select at least one argument span and its stance (supporting/opposing); alternatively, if they could not find an argument span, they had to solve a simple math problem. We employed two quality control measures: a qualification test and periodic attention checks.[11] On an initial batch of 160 sentences, we collected votes from nine workers. To determine the optimal number of workers for the final study, we did majority voting on the token level (ties broken as non-arguments) for both the expert study and workers from the initial crowd study. We artificially reduced the number of workers (1-9) and calculated percentage overlap averaged across all worker combinations (for worker numbers ≤ 9). Whereas the overlap was highest with 80.2% at nine votes, it only dropped to 79.5% for five votes (and decreased more significantly for fewer votes). We deemed five votes to be an acceptable tradeoff between quality and cost. The agreement between experts and crowd in the five-worker setup is $\alpha_{u_{nom}}$ = 0.71, which is substantial (Landis and Koch 1977).

The final **gold standard** labels on the 8000 sampled sentences were determined using a variant of Bayesian Classifier Combination (Kim and Ghahramani 2012), referred to as IBCC in Simpson and Gurevych (2019)'s modular framework for Bayesian aggregation of sequence labels. This method has been shown to yield results superior to majority voting or MACE (Hovy et al. 2013). After manual inspection of the resulting gold standard, we merged all consecutive segments of the same stance, to form a gold standard with coherent segments separated by at least one other token label (most of the time NON).

3.5 Dataset Splits

We create two different dataset splits. (i) An in-domain split. This lets us evaluate how models perform on known vocabulary and data distributions. (ii) A cross-domain split. This lets us evaluate how well a model generalizes for unseen topics and distributions different from the training set.[12] In the cross-domain setup, we defined topics T1-T5 to be in the train set, topic T6 in the dev set and topics T7 and T8 in the test set. For the in-domain setup, we excluded topics T7 and T8 (cross-domain test set), and used the first 70% of the topics T1-T6 for train, the next 10% for dev and the remaining 20% for test. The samples from the in-domain test set were also excluded in the cross-domain train and dev sets. As a result, there are 4000 samples in train, 800 in dev and 2000 in test for the cross-domain split; and 4200 samples in train, 600 in dev and 1200 in test for the in-domain split. Our definition of the two splits guarantees that train/dev sets (in-domain or cross-domain) do not overlap with test sets. The assignment of sentences to the two splits is released as part of AURC-8.

3.6 Dataset Statistics

The resulting dataset, AURC-8, consists of 8000 annotated sentences; 3500 (43.8%) of which are non-argumentative. The 4500 argumentative sentences are divided into 1951

[10] Strict adherence to grammatical correctness would require that all spans are clauses. A reviewer asked if there are examples of non-clauses in our data. Although these cases are rare in our gold standard, they do occur. Examples include noun phrases like "risky animal experiments" (against the topic "cloning") and clauses that are missing the subject, which is usually to be equated with the topic, e.g., "have no real impact on improving student achievement" (for the topic "school uniforms"). These argument units are easily comprehended by humans even though they are not complete clauses in the strict grammatical sense.

[11] Workers had to be located in the US, CA, AU, NZ or UK, with an acceptance rate of 95% or higher. Payment was $0.42 per HIT, corresponding to US federal minimum wage ($7.25/hour).

[12] We use *cross-domain* rather than *cross-topic* (Stab et al. 2018b) here, as the former is the more common term.

(43.4%) single PRO argument sentences, 1799 (40.0%) single CON argument sentences and 750 (16.7%) sentences that are many possible combinations of PRO and CON arguments with up to five single argument units in a sentence. The total number of argumentative segments is 4973. Thus, due to the token-level annotation the number of arguments is higher by 10.5% compared to what it would be with a sentence-level approach (4500). If we propagate the label of a sentence to all its tokens, then 100% of tokens of an argumentative sentence are argumentative. This ratio drops to 69.9% in our token-level setup, reducing the amount of non-argumentative tokens otherwise incorrectly labeled as argumentative in a sentence.

4 Experimental Setup

We train and evaluate in two different setups: token-level and sentence-level. In the *token-level setup*, models are trained and evaluated on the gold standard as is. For the *sentence-level setup*, we use a sentence-level gold standard that is modified by converting token-level gold standard or predictions to a prediction for the sentence as follows. If only NON occurs, the sentence is labeled NON. If NON and only PRO (resp. only CON) occurs, PRO (resp. CON) is chosen. If both PRO and CON occur, the more frequent label of the two is assigned. In the few exceptional cases where PRO and CON are equally frequent, one of them is chosen by chance.

4.1 Evaluation Measures

We compute three different evaluation measures of AURC: **token** F_1, **segment** F_1 and **sentence** F_1.

Token F_1 is defined as the average of the three class F_1's for PRO, CON and NON, each computed for all tokens in the evaluation set. This is the core evaluation measure of our work since our goal is argument retrieval at a more fine-grained level than prior work.[13]

To compute **segment** F_1 for a sentence, we look at all pairs <gold segment g, predicted segment p> and compute:

$$r = \frac{|g \cap p|}{\max(|g|, |p|)}$$

If $r > 0.5$, then we count p as a true positive if the labels are the same. We do this only for PRO and CON segments, so the correct recognition of NON segments (which is of less benefit in an argument retrieval scenario) is not rewarded. However, if the sentence contains no PRO or CON span and no PRO or CON span is predicted, then we set F_1 for this sentence to 1.0. Thus, an AURC model is rewarded for recognizing a sentence that is entirely NON. Macro-averaged segment F_1 is then the average of segment F_1's for all sentences in the evaluation set.

Segment F_1 is a measure of evaluating fine-grained argument mining that is complementary to token F_1. It is more forgiving than token F_1 since segment boundaries do not have to be recognized exactly. At the same time, it punishes cases where tokens are mostly recognized correctly, but the segments that a downstream application may need (e.g., a user interface displaying arguments) are not. For example, a gold segment may be split in half by a single NON token, resulting in two incomplete predicted segments.

For **sentence** F_1, a single label is predicted for each sentence. The three class F_1's for PRO, CON and NON are computed for the sentences in the evaluation set and averaged to yield the overall sentence F_1. Gold standard labels for sentences are produced as described above (*sentence-level setup*).

4.2 Methods

We test and extend models that have recently well performed on AM and sequence labeling (Reimers et al. 2019).

Baseline. The majority baseline labels each token of the input sentence with NON, the most frequent class.

FLAIR. We use FLAIR (Akbik, Blythe, and Vollgraf 2018), a recent neural sequence labeling library with state-of-the-art performance on many sequence labeling tasks. For the token-level setup, we use the FLAIR SequenceTagger with a CRF and with stacked character embeddings (Lample et al. 2016). For the sentence-level setup, we use the FLAIR TextClassifier. In both cases, the selected model is a BiLSTM (Hochreiter and Schmidhuber 1997) and we use GloVe word embeddings (Pennington, Socher, and Manning 2014) and FlairEmbeddings (provided by the library).

BERT. We include two pretrained models of BERT[14] (Devlin et al. 2019) as a recent state-of-the-art model that achieves impressive results on many tasks including sequence labeling: $\text{BERT}_{\text{BASE}}$ (cased, base) and $\text{BERT}_{\text{LARGE}}$ (cased, large, with whole-word-masking). A third model adds a CRF (Lafferty, McCallum, and Pereira 2001) as a task specific sequence labeling layer: $\text{BERT}_{\text{LARGE}}$+CRF. We finetune BERT's parameters (including the CRF) for the AURC task. Since we work in the information-seeking paradigm of AM, the topic (i.e., the search query) is of key importance. A span of words that is a supporting argument (PRO) for "nuclear energy" will generally be NON for the topic "gun control". Thus we provide BERT with an additional input, the topic, separated by a [SEP] token from the input sequence. We found that this additional topic information was consistently helpful for AURC. Token- and sentence-level setups for the BERT models only differ with regard to gold standard data.

5 Results

Table 2 shows AURC results for our five methods (Section 4.2) for in-domain (lines 4–8) and cross-domain (9–13). The human upper bound (i.e., the performance on AURC by the three graduate employees) is given on line 14. We report the mean of three runs with different random seeds for non-deterministic methods (Reimers and Gurevych 2017). Information about the hyperparameters, precision and recall, as well as separat evaluation for unit recognition (2 classes) and classification (3 classes) can be found in Section A and Section B of the appendix.

[13] To illustrate this point, we use the term "argument retrieval" rather than "information-seeking AM" in the following.

[14] https://github.com/huggingface/pytorch-pretrained-bert

		model	token F_1				segment F_1				sentence F_1			
			token-level setup		sentence-level setup		token-level setup		sentence-level setup		token-level setup		sentence-level setup	
			dev	test	dev	test	dev	test	dev	test	dev	test	dev	test
in-domain	4	majority baseline	.258	.254	.258	.254	.478	.463	.478	.463	.216	.211	.216	.211
	5	FLAIR	.642	.613	.442	.473	.661	.623	.473	.492	.651	.620	.495	.523
	6	$\text{BERT}_{\text{BASE}}$	.702	.654	.591	.581	.666	.628	**.615**	**.601**	.717	.673	.680	.665
	7	$\text{BERT}_{\text{LARGE}}$	.732	.683	**.671**	**.627**	.749	.709	.599	.567	.738	.709	**.759**	.715
	8	$\text{BERT}_{\text{LARGE}}$+CRF	**.743**	**.696**	.637	.622	**.750**	**.724**	.552	.547	**.744**	**.711**	.731	**.725**
cross-domain	9	majority baseline	.245	.240	.245	.240	.394	.379	.394	.379	.188	.183	.188	.183
	10	FLAIR	.419	.433	.388	.401	.457	.458	.402	.401	.386	.402	.428	.418
	11	$\text{BERT}_{\text{BASE}}$	.554	.563	.431	.474	.504	.508	.445	**.473**	.550	.556	.473	.510
	12	$\text{BERT}_{\text{LARGE}}$	.604	.596	**.550**	**.544**	.653	.626	**.487**	**.473**	.606	.598	**.628**	**.602**
	13	$\text{BERT}_{\text{LARGE}}$+CRF	**.615**	**.620**	.505	.519	**.681**	**.649**	.456	.464	**.627**	**.610**	.569	.573
	14	human performance	.763				.709				.799			

Table 2: Token F_1, segment F_1 and sentence F_1 on AURC-8 dev and test. Bold: best performance per column and split (in-domain, cross-domain).

5.1 Model Comparison

The BERT models always perform better than the majority baseline and FLAIR (6–8 vs. 4–5 and 11–13 vs. 9–10). For sentence-level setup and segment F_1, the majority baseline is close or even better than FLAIR (lines 4 vs. 5 and 9 vs. 10). The reason for the high performance of the majority baseline for segment F_1 (lines 4 & 9) is that about 40% of segments are NON segments – so a method that labels everything as NON will get a high score for segment identification. $\text{BERT}_{\text{BASE}}$ mostly performs worse than $\text{BERT}_{\text{LARGE}}$ (lines 6 vs. 7–8 and 11 vs. 12–13), but still at an acceptable level. Hence, $\text{BERT}_{\text{BASE}}$, which requires only one GPU for training, is a good option if computational resources are limited.

5.2 Token- vs. Sentence-Level Setup

$\text{BERT}_{\text{LARGE}}$+CRF is consistently the best model for token-level setup (bolded numbers in Table 2), whereas $\text{BERT}_{\text{LARGE}}$ is the best model for sentence-level setup (with one exception: for segment F_1 in-domain, $\text{BERT}_{\text{BASE}}$ is better: line 6). Thus, for the token-level setup, the CRF layer (lines 8 & 13) always performs better than imposing no sequence constraints (lines 7 & 12) – whereas the sequence constraint is not beneficial for the sentence-level setup.

The difference between the two setups is noticeable for token and segment F_1, but smaller for sentence F_1. As one would expect, the token-level setup is not a good match for sentence-level evaluation (for both in- and cross-domain experiments).

5.3 In-Domain vs. Cross-Domain

The best in-domain BERT numbers are pretty high, with segment F_1 (line 8) even above human performance (line 14), while for token and sentence F_1 the gap to human performance is between 7 and 9 percentage points. Cross-domain is clearly more challenging since models are evaluated on unseen topics. Here, F_1 scores drop to between 0.61 and 0.65 for the token-level models. Training on only 5 topics (see Section 3.5) gives insufficient information for the transfer to unseen topics. These numbers are, however, in line with the current state-of-the-art for sentence-level AM; see Reimers et al. (2019) who report an F_1 score of 0.63 for a similar architecture but slightly different experimental setup with 7 training topics (cf. Stab et al. (2018b)). Given that the segment F_1 score (line 13) is only 6 percentage points below human agreement, we expect that in cases where exact argument segment boundaries are not of high importance, cross-domain transfer will still work fine.

6 Analysis

We now discuss the main errors our models make (Section 6.1) and investigate robustness against incorrectly recognized sentence boundaries (Section 6.2).

6.1 Recognition and Classification Errors

The main types of errors we observed were: (i) the *span* of an argumentative segment is not correctly recognized and (ii) the *stance* is not correctly classified.

Span. The first major type of error related to spans is that the beginning and/or end of a segment is incorrectly recognized. We analyzed the predictions by the best token-level model (line 8 & 13 in Table 2) on dev. The average segment length (in tokens) is 17.5 for gold and 16.4 for predicted (in-domain) and 17.3 for gold and 15.1 for predicted (cross-domain); thus, predicted segments are on average shorter than gold segments. However, this will often be benign in an application since annotators tended to include boundary words that are not strictly necessary.

The second major type of error related to spans is that a segment is broken into several segments or several segments are merged into one segment. Our models predict 7.6% more segments in-domain and 13.1% more segments cross-domain than there are in the gold standard. Shorter spans in the predictions (which result in more spans) are mostly due to segment splits on function words like “and”, “that” and “yet”. Figure 2 gives examples of span errors for

	model	topic	sentence
1	gold standard	school uniforms	It has been argued that uniforms force conformity , yet many uniform - wearing students like the fact that everyone looks the same .
2	BERT$_{\text{LARGE}}$	school uniforms	It has been argued that uniforms force conformity , yet many uniform - wearing students like the fact that everyone looks the same .
3	BERT$_{\text{LARGE}}$+CRF	school uniforms	It has been argued that uniforms force conformity , yet many uniform - wearing students like the fact that everyone looks the same .
4	gold standard	abortion	Women who want abortions will get them , even if they have to risk their lives to get them .
5	BERT$_{\text{LARGE}}$	abortion	Women who want abortions will get them , even if they have to risk their lives to get them .
6	BERT$_{\text{LARGE}}$+CRF	abortion	Women who want abortions will get them , even if they have to risk their lives to get them .
7	gold standard	school uniforms	While they may create a positive sense of unity , they can also imply the sacrifice of individuality to a group mentality .
8	BERT$_{\text{LARGE}}$	school uniforms	While they may create a positive sense of unity , they can also imply the sacrifice of individuality to a group mentality .
9	BERT$_{\text{LARGE}}$+CRF	school uniforms	While they may create a positive sense of unity , they can also imply the sacrifice of individuality to a group mentality .

Figure 2: Gold standard segments and segments predicted by BERT$_{\text{LARGE}}$ and BERT$_{\text{LARGE}}$+CRF for three example sentences. Green: PRO. Red: CON.

the model without CRF (lines 2, 5 & 8). The model with CRF (lines 3, 6 & 8) correctly recognizes beginning and end of the span more often, while the one without CRF tends to split segments (line 5) and sometimes even creates one-word or two-word segments (line 8). So the addition of a CRF layer improves segmentation – no overly short segments, fewer segments incorrectly broken up – and therefore results in the best overall performance in our experiments.

Stance. The classification of stance is as important for argument retrieval as span detection. The model with CRF sometimes assigns the same stance for all segments within a sequence – see the examples shown in Figure 2 (lines 3 & 9). This may be due to only a few mixed segments (both PRO and CON stance) appearing in the training set (43 for in-domain and 48 for cross-domain). We plan to incorporate additional stance focused information – such as sentiment – in future work to improve stance classification. Another possible improvement could be achieved with multi-task training. There are additional sentence-level datasets for stance detection (Mohammad et al. 2016) available. In a multi-task setup, we can exploit these to improve stance detection for our task.

More detailed analysis of the subtask of recognition and classification (see Table 4 in Section B of the appendix) showed that when combining PRO and CON labels into one ARG label for the best token-level system (BERT$_{\text{LARGE}}$+CRF) yields a smaller drop in performance between in- and cross-domain. Consequently, our best token-level model manages cross-domain transfer very well for argument unit recognition, whereas argument (stance) classification remains challenging in a cross-domain setup.

6.2 Sentence Boundaries

One of the main benefits of a token-level argument mining approach is that it is robust against errors in sentence boundary detection. We now analyze how our token-level models perform when they are not given any sentence boundary information. We perform this experiment with our best models (BERT$_{\text{LARGE}}$+CRF for token-level setup and BERT$_{\text{LARGE}}$ for sentence-level setup).

			token-level setup		sentence-level setup	
			dev	test	dev	test
in-domain	3	token F_1	**.671**	**.642**	.507	.510
	4	segment F_1	**.634**	**.592**	.254	.235
	5	sentence F_1	**.514**	.504	.510	**.509**
cross-domain	6	token F_1	**.573**	**.562**	.442	.452
	7	segment F_1	**.554**	**.538**	.175	.154
	8	sentence F_1	.433	.449	**.448**	**.473**

Table 3: Token-level vs. sentence-level setup for input sequences that cross sentence boundaries (i.e., no sentence segmentation). Bold: best performance per line and dev/test

We create a new dataset by concatenating, for each topic, all sentences of that topic. The resulting document is then processed by moving the model over it using a sliding window of size 45 and a stepsize of 1. In this regime, most input sequences do not correspond to sentences, but instead contain pieces of several sentences. Table 3 shows results.

We see that the token-level setup performs much better on input without sentence boundaries than the sentence-level setup – both for token F_1 and for segment F_1, both in-domain and cross-domain (lines 3-4, 6-7). While the performance is clearly lower than in Table 2, it declines gracefully for the token-level setup whereas the drop in performance for the sentence-level setup is large, especially for segment F_1; e.g., on the test set: .154 (line 7, Table 3) vs. .473 (line 12, Table 2). This clearly indicates the robustness of our approach against errors in sentence boundary detection.

7 Conclusion

We define the task of fine-grained argument unit recognition and classification (AURC), and release the benchmark AURC-8 for this task. Our work is the first to show that information-seeking AM can be applied on very heterogeneous data (a web crawl in our case) on the level of tokens. We demonstrated that AURC-8 has good quality in terms of annotator agreement: the required annotations can be crowdsourced using specific data selection and filtering methods combined with a slot filling approach. We test and adapt several sequence tagging methods for AURC, achieving very good performance on known domains. As in previous work on sentence-level argument retrieval, the methods suffer a drop in performance on unknown domains. Performance on unknown domains is, however, quite good when segment boundaries do not need to be determined exactly.

We have also shown that our approach, as opposed to sentence-level argument retrieval, does not depend on correct sentence boundaries. By making the analysis more fine-grained, we increase the recall of arguments and make their representation more accurate for the user as well as for downstream tasks. If required by downstream applications, the template defined in Section 3.4 can be used to reconstruct complete arguments for most of AURC-8. For the example in Figure 2 line 7, we would get the following to statements: "School uniforms should be supported because they may create a sense of positive unity" and "School uniforms should be opposed because they can also imply the sacrifice of individuality to a group mentally."

This work makes possible a range of potential follow-up investigations. Since our approach does not depend on correct sentence boundaries, it is well-suited for noisy data from social media or (continuous) speech data, e.g., political debates. Furthermore, the benefit of our fine-grained argument annotations should be explored in downstream applications like claim validation and fact checking, which rely on decomposed evidence. Finally, our retrieval and annotation pipeline can be used to annotate argument units for additional topics, thus enabling improved domain transfer.

Acknowledgments

We gratefully acknowledge support by Deutsche Forschungsgemeinschaft (DFG) (SPP-1999 Robust Argumentation Machines (RATIO), SCHU2246/13), as well as by the German Federal Ministry of Education and Research (BMBF) under the promotional reference 03VP02540 (ArgumenText).

A Hyperparameters

This section lists the hyperparameters used for the experimental systems described in the main part of the paper.

For FLAIR in the token-level model, we used a learning rate of 1e-1 with gradual decreasing, hiddensize=256 and for the sentence-level model the same setting for the learning rate, but with hiddensize=512.

For $\text{BERT}_{\text{LARGE}}$ and $\text{BERT}_{\text{LARGE}}$+CRF, we used the large cased pretrained model with whole word masking and in the token-level setup a learning rate of 1e-5 for in- and cross-domain. For the CRF we used the [CLS] token as the START and [SEP] as the END token, so we considered only the sequence input (without the topic) for this setup. For the evaluation we considered only the predicted tag sequence between the START and END token.

We kept the learning rate at 4e-5 for the sentence-level $\text{BERT}_{\text{LARGE}}$ model and at 1e-5 for the $\text{BERT}_{\text{LARGE}}$+CRF and used the AdamW optimizer. The max. length of the tokenized BERT input was set to 64 tokens and we always had a dropout rate of 0.1.

All experiments were run three times with different seeds, a trainings batch size of 32 and for a max. of 100 epochs, with earlier stopping if the performance/loss did not improve/decreased significantly (after ten epochs).

		domain	set	precision	recall	F_1
classific. (3 classes)	2	in-domain	dev	.742	.745	.743
	3	cross-domain	dev	.632	.638	.615
	4	in-domain	test	.717	.681	.696
	5	cross-domain	test	.637	.609	.620
recogn. (2 classes)	6	in-domain	dev	.812	.814	.813
	7	cross-domain	dev	.805	.792	.797
	8	in-domain	test	.793	.776	.782
	9	cross-domain	test	.782	.767	.770

Table 4: Token-level precision, recall and macro F_1 results for argument classification (PRO, CON, NON) and recognition (ARG, NON) for the prediction of the best token-level model.

B Additional Information

Table 4 provides detailed information about token-level precision, recall and macro F_1 scores and the separate evaluation on three classes (PRO vs. CON. vs. NON) and two classes (ARG vs. NON), where PRO and CON labels were converted into a single ARG label.

References

Aharoni, E.; Polnarov, A.; Lavee, T.; Hershcovich, D.; Levy, R.; Rinott, R.; Gutfreund, D.; and Slonim, N. 2014. A benchmark dataset for automatic detection of claims and evidence in the context of controversial topics. In *Argumentation Mining Workshop*, 64–68.

Ajjour, Y.; Chen, W.-F.; Kiesel, J.; Wachsmuth, H.; and Stein, B. 2017. Unit segmentation of argumentative texts. In *Argument Mining Workshop*, 118–128.

Akbik, A.; Blythe, D.; and Vollgraf, R. 2018. Contextual string embeddings for sequence labeling. In *COLING'18*, 1638–1649.

Cabrio, E., and Villata, S. 2018. Five years of argument mining: a data-driven analysis. In *IJCAI'18*, 5427–5433.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL'19*, 4171–4186.

Eckle-Kohler, J.; Kluge, R.; and Gurevych, I. 2015. On the role of discourse markers for discriminating claims and premises in argumentative discourse. In *EMNLP'15*, 2236–2242.

Habernal, I., and Gurevych, I. 2017. Argumentation mining in user-generated web discourse. *Computational Linguistics* 43(1):125–179.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Hovy, D.; Berg-Kirkpatrick, T.; Vaswani, A.; and Hovy, E. 2013. Learning whom to trust with mace. In *NAACL'13*, 1120–1130.

Hua, X., and Wang, L. 2017. Understanding and detecting supporting arguments of diverse types. In *ACL'17*, 203–208.

Kim, H.-C., and Ghahramani, Z. 2012. Bayesian classifier combination. In *AISTATS'12*, 619–627.

Krippendorff, K.; Mathet, Y.; Bouvry, S.; and Widlöcher, A. 2016. On the reliability of unitizing textual continua: Further developments. *Quality & Quantity* 50(6):2347–2364.

Kuribayashi, T.; Ouchi, H.; Inoue, N.; Reisert, P.; Miyoshi, T.; Suzuki, J.; and Inui, K. 2019. An empirical study of span representations in argumentation structure parsing. In *ACL'19*, 4691–4698.

Lafferty, J. D.; McCallum, A.; and Pereira, F. C. N. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML'01*, 282–289.

Lample, G.; Ballesteros, M.; Subramanian, S.; Kawakami, K.; and Dyer, C. 2016. Neural architectures for named entity recognition. In *NAACL'16*, 260–270.

Landis, J. R., and Koch, G. G. 1977. The measurement of observer agreement for categorical data. *Biometrics* 33(1):159–174.

Levy, R.; Bilu, Y.; Hershcovich, D.; Aharoni, E.; and Slonim, N. 2014. Context dependent claim detection. In *COLING'14*, 1489–1500.

Li, M.; Geng, S.; Gao, Y.; Peng, S.; Liu, H.; and Wang, H. 2017. Crowdsourcing argumentation structures in Chinese hotel reviews. In *IEEE Int. Conference on Systems, Man, and Cybernetics*, 87–92.

Lippi, M., and Torroni, P. 2016. Argument mining from speech: Detecting claims in political debates. In *AAAI'16*, 2979–2985.

Miller, T.; Sukhareva, M.; and Gurevych, I. 2019. A streamlined method for sourcing discourse-level argumentation annotations from the crowd. In *NAACL'19*, 1790–1796.

Mohammad, S.; Kiritchenko, S.; Sobhani, P.; Zhu, X.; and Cherry, C. 2016. SemEval-2016 task 6: Detecting stance in tweets. In *SemEval'16*, 31–41.

Nguyen, H. V., and Litman, D. J. 2018. Argument mining for improving the automated scoring of persuasive essays. In *AAAI'18*, 5892–5899.

Palau, R. M., and Moens, M.-F. 2009. Argumentation mining: The detection, classification and structure of arguments in text. In *ICAIL'09*, 98–107.

Peldszus, A., and Stede, M. 2013. From argument diagrams to argumentation mining in texts: A survey. *Int. J. Cogn. Inform. Nat. Intell.* 7(1):1–31.

Peldszus, A., and Stede, M. 2015. Joint prediction in MST-style discourse parsing for argumentation mining. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 938–948. Lisbon, Portugal: Association for Computational Linguistics.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *EMNLP'14*, 1532–1543.

Reimers, N., and Gurevych, I. 2017. Reporting score distributions makes a difference: Performance study of lstm-networks for sequence tagging. In *EMNLP'17*, 338–348.

Reimers, N.; Schiller, B.; Beck, T.; Daxenberger, J.; Stab, C.; and Gurevych, I. 2019. Classification and clustering of arguments with contextualized word embeddings. In *ACL'19*, 567–578.

Reisert, P.; Inoue, N.; Kuribayashi, T.; and Inui, K. 2018. Feasible annotation scheme for capturing policy argument reasoning using argument templates. In *Argument Mining Workshop*, 79–89.

Shnarch, E.; Alzate, C.; Dankin, L.; Gleize, M.; Hou, Y.; Choshen, L.; Aharonov, R.; and Slonim, N. 2018. Will it blend? blending weak and strong labeled data in a neural network for argumentation mining. In *ACL'18*, 599–605.

Simpson, E. D., and Gurevych, I. 2019. A Bayesian approach for sequence tagging with crowds. In *EMNLP-IJCNLP'19*, 1093–1104.

Stab, C., and Gurevych, I. 2014. Annotating argument components and relations in persuasive essays. In *COLING'14*, 1501–1510.

Stab, C.; Daxenberger, J.; Stahlhut, C.; Miller, T.; Schiller, B.; Tauchmann, C.; Eger, S.; and Gurevych, I. 2018a. Argumentext: Searching for arguments in heterogeneous sources. In *NAACL'18*, 21–25.

Stab, C.; Miller, T.; Schiller, B.; Rai, P.; and Gurevych, I. 2018b. Cross-topic argument mining from heterogeneous sources. In *EMNLP'18*, 3664–3674.

Wachsmuth, H.; Potthast, M.; Al Khatib, K.; Ajjour, Y.; Puschmann, J.; Qu, J.; Dorsch, J.; Morari, V.; Bevendorff, J.; and Stein, B. 2017. Building an argument search engine for the web. In *Argument Mining Workshop*, 49–59.

Walton, D., and Gordon, T. F. 2017. Argument Invention with the Carneades Argumentation System. *SCRIPTed* 14(2):168–207.

Wyner, A.; Mochales-Palau, R.; Moens, M.-F.; and Milward, D. 2010. Approaches to text mining arguments from legal cases. In *Semantic Processing of Legal Texts: Where the Language of Law Meets the Law of Language*. Springer Berlin Heidelberg. 60–79.